# Supplementary Material for Tailoring Differential Privacy to Any Explainer

This is the supplementary material for the paper "Tailoring Differential Privacy to Any Explainer". Section I contains proofs for theorems, lemmas and corollaries used in the main body of the paper. We apply the DP enabler theorem to Partial Dependence Plots [1] in Section II. Here, we also verify the applicability of Generic DP Plots and Generic DP Rank Aggregation to one further explainer each. Section III contains the evaluation of the new explainer-specific design for Partial Dependence Plots and additional results for DP ALE and Generic DP ALE.

## I. Proofs

### A. Proof that DP Plots is DP

**Lamma IV.1.** Algorithm 3 is DP with privacy budget $\varepsilon$.

*Proof.* By using average as the aggregation function, we bound the sensitivity of the output to $m \cdot (max_y - min_y)/l$. Namely, only one subset $O^{(i)}$ changes when one record changes between $D$ and $D'$. Thus, the calculation of $y$ values becomes differentially private with privacy budget $\varepsilon$ by adding Laplacian noise with the required scale $\sigma = m \cdot (max_y - min_y)/(l \cdot \varepsilon)$ in Line 12 of Algorithm 3. No noise needs to be added to the $x$ values since they only depend on the given bounds $min_x$, $max_x$ and the resolution $m$, not on $D$. Since the final output only consists of the $x$ and $y$ values, Algorithm 3 is DP with budget $\varepsilon$. □

### B. Proof of Sensitivity for Generic Rank Aggregation

First, we require some preliminary definitions and corollaries. Let there be $d$ candidates. Let $r$ be a vector of the number of votes for each candidate. Let there $m$ ballots $b \in B$ with the data set split between them (for their calculation). A ballot is a permutation of $\{1, \dots, d\}$ as a vector. These are the points the ballot allocates to each of the candidates. If an entry changes in the data set, only one ballot $b \in B$ will be affected. W.l.o.g. let that ballot be $b^{(k)}$.

$$\begin{aligned}
\Delta r &= \max_{D,D'} \|r_D - r_{D'}\|_1 \\
&= \max_{D,D'} \|(\sum_{j=1}^{m} b_i^{(j)})_{i=1}^{d} - (\sum_{j=1}^{m} \overline{b}_i^{(j)})_{i=1}^{d}\|_1 \\
&= \max_{D,D'} \|(\sum_{j=1}^{m} b_i^{(j)} - \overline{b}_i^{(j)})_{i=1}^{d}\|_1 \\
&= \max_{D,D'} \|(b_i^{(k)} - \overline{b}_i^{(k)})_{i=1}^{d}\|_1 \\
&= \max_{D,D'} \sum_{i=1}^{d} |b_i^{(k)} - \overline{b}_i^{(k)}|
\end{aligned} \tag{1}$$

The sensitivity of the borda vote is the maximum of this sum of absolute differences of two permutations over $\{0, \dots d\}$. We can consider one permutation as fixed at $(0, \dots, d)$ and only consider what form the second permutation $\phi$ must take to maximize the following function:

$$\mathcal{C}(\phi) := \sum_{i=0}^{d} |i - \phi(i)| \tag{2}$$

**Corollary I.0.1.** Let $\phi$ be a permutation over $\{0, \dots, d\}$ with

$$\exists i \in \{0, \dots, d\} : i \lesseqgtr \frac{d}{2} \text{ and } \phi(i) \lesseqgtr \frac{d}{2}.$$

Then exists $\phi'$ with

$$\mathcal{C}(\phi') > \mathcal{C}(\phi).$$

*Proof.* Consider the case $i < \frac{d}{2}$ and $\phi(i) < \frac{d}{2}$. The other case can be proven analogously. There exists $j \geq \frac{d}{2}$ with $\phi(j) \geq \frac{d}{2}$ because of the pigeon hole principle. We construct a new permutation $\phi'$ with

$$\phi'(k) = \begin{cases} \phi(j) & k = i \\ \phi(i) & k = j \\ \phi(k) & \text{otherwise} \end{cases}$$

It holds

$$\begin{aligned}\mathcal{C}(\phi') - \mathcal{C}(\phi) &= \sum_{k=0}^{d} |k - \phi'(k)| - \sum_{k=0}^{d} |k - \phi(k)| \\ &= |i - \phi'(i)| - |i - \phi(i)| + |j - \phi'(j)| \\ &\quad - |j - \phi(j)| \\ &= |i - \phi(j)| - |i - \phi(i)| + |j - \phi(i)| \\ &\quad - |j - \phi(j)| \\ &= \phi(j) - i + j - \phi(i) - |i - \phi(i)| - |j - \phi(j)| \\ &= *\end{aligned} \tag{3}$$

**Case 1:** $i \geq \phi(i)$ **and** $j \geq \phi(j)$**.**
$* = \phi(j) - i + j - \phi(i) - i + \phi(i) - j + \phi(j) = 2(\phi(j) - i) > 2(\frac{d}{2} - \frac{d}{2}) = 0$

**Case 2:** $i \geq \phi(i)$ **and** $j < \phi(j)$**.**
$* = \phi(j) - i + j - \phi(i) - i + \phi(i) + j - \phi(j) = 2(j - i) > 2(\frac{d}{2} - \frac{d}{2}) = 0$

**Case 3:** $i < \phi(i)$ **and** $j \geq \phi(j)$**.**
$* = \phi(j) - i + j - \phi(i) + i - \phi(i) - j + \phi(j) = 2(\phi(j) - \phi(i)) > 2(\frac{d}{2} - \frac{d}{2}) = 0$

**Case 4:** $i < \phi(i)$ **and** $j < \phi(j)$**.**
$* = \phi(j) - i + j - \phi(i) + i - \phi(i) + j - \phi(j) = 2(j - \phi(i)) > 2(\frac{d}{2} - \frac{d}{2}) = 0$

Thus $\mathcal{C}(\phi') > \mathcal{C}(\phi)$. □

**Definition I.1.** A permutation $\phi$ over $\{0, \ldots, d\}$ is called *specially ordered* iff it fulfills the following two conditions:

$$\forall i \in \{0, \ldots, d\} : i < \frac{d}{2} \Rightarrow \phi(i) \geq \frac{d}{2}$$

and

$$\forall i \in \{0, \ldots, d\} : i > \frac{d}{2} \Rightarrow \phi(i) \leq \frac{d}{2}$$

**Corollary I.0.2.** Let $\phi, \phi'$ be two specially ordered permutations over $\{0, \ldots, d\}$. Then

$$\mathcal{C}(\phi) = \mathcal{C}(\phi').$$

*Proof.* Permutation $\phi$ can be changed step-wise to $\phi'$ by switching two indices in each step. We may switch $i, j < \frac{d}{2}$ (both in the lower half) or $i, j > \frac{d}{2}$ (both in the upper half) or, if needed, $\phi(\frac{d}{2}) = \frac{d}{2}$ to any other index. These switches ensure that we always stay specially-ordered and can reach any specially-ordered permutation. We show that all such operations do not change the value of $\mathcal{C}$.

**Case 1: Switching indices in one half does not change $\mathcal{C}$.**
Let $\varphi$ be a specially ordered permutation. Let $\varphi'$ be a permutation with

$$\varphi'(k) = \begin{cases} \varphi(j) & k = i < \frac{d}{2} \\ \varphi(i) & k = j < \frac{d}{2} \\ \varphi(k) & \text{otherwise.} \end{cases}$$

Observe that $\varphi'$ must also be specially ordered. The following holds with the specially-ordered-property of the two permutations.

$$\begin{aligned}\mathcal{C}(\varphi') - \mathcal{C}(\varphi) &= \sum_{k=0}^{d} |k - \varphi'(k)| - \sum_{k=0}^{d} |k - \varphi(k)| \\ &= |i - \varphi'(i)| - |i - \varphi(i)| + |j - \varphi'(j)| \\ &\quad - |j - \varphi(j)| \\ &= |i - \varphi(j)| - |i - \varphi(i)| + |j - \varphi(i)| \\ &\quad - |j - \varphi(j)| \\ &= \varphi(j) - i - \varphi(i) + i + \varphi(i) - j \\ &\quad - \varphi(j) + j \\ &= 0\end{aligned} \tag{4}$$

Thus $\mathcal{C}(\varphi') = \mathcal{C}(\varphi)$. Analogous for $i, j > \frac{d}{2}$.

**Case 2: Switching $\frac{d}{2}$ away from or to index $\frac{d}{2}$ does not change $\mathcal{C}$.** This case only applies if $d$ is even. Let $\varphi$ be a specially ordered permutation with $\varphi(\frac{d}{2}) = \frac{d}{2}$ and let $\varphi'$ be a permutation with

$$\varphi'(k) = \begin{cases} \varphi(\frac{d}{2}) = \frac{d}{2} & k = i \\ \varphi(i) & k = \frac{d}{2} \\ \varphi(k) & \text{otherwise.} \end{cases}$$

Observe that $\varphi'$ must also be specially ordered. The following holds with the specially-ordered-property of the two permutations.

$$\begin{aligned}\mathcal{C}(\varphi') - \mathcal{C}(\varphi) &= \sum_{k=0}^{d} |k - \varphi'(k)| - \sum_{k=0}^{d} |k - \varphi(k)| \\ &= |i - \varphi'(i)| - |i - \varphi(i)| + |\frac{d}{2} - \varphi'(\frac{d}{2})| \\ &\quad - |\frac{d}{2} - \varphi(\frac{d}{2})| \\ &= |i - \varphi(\frac{d}{2})| - |i - \varphi(i)| + |\frac{d}{2} - \varphi(i)| \\ &\quad - |\frac{d}{2} - \varphi(\frac{d}{2})| \\ &= |i - \frac{d}{2}| - |i - \varphi(i)| + |\frac{d}{2} - \varphi(i)| \\ &= *\end{aligned} \tag{5}$$

**Case 2.1** $i < \frac{d}{2}$**:** $* = \frac{d}{2} - i + \varphi(i) - \frac{d}{2} - \varphi(i) + i = 0$

**Case 2.2** $i \geq \frac{d}{2}$**:** $* = i - \frac{d}{2} + \frac{d}{2} - \varphi(i) - i + \varphi(i) = 0$

Thus $\mathcal{C}(\varphi') = \mathcal{C}(\varphi)$.

In each step of the transformation from $\phi$ to $\phi'$, $\mathcal{C}$ does not change. Therefore, $\mathcal{C}(\phi) = \mathcal{C}(\phi')$. □

**Corollary I.0.3.** Let $\phi$ be a specially ordered permutation over $\{0, \ldots, d\}$. Then

$$\mathcal{C}(\phi) = \max_{\phi^*} \mathcal{C}(\phi^*).$$

*Proof.* Let us assume that a non-specially-ordered permutation $\phi$ maximizes $\mathcal{C}$. According to Corollary I.0.1, there must exist

a permutation $\phi'$ with $\mathcal{C}(\phi') > \mathcal{C}(\phi)$. This contradicts that $\phi$ maximizes $\mathcal{C}$. Therefore, our assumption was wrong and a specially-ordered permutation must maximize $\mathcal{C}$. With Corollary I.0.2, it is clear that all specially-ordered permutations maximize $\mathcal{C}$ if one specially-ordered permutation maximizes $\mathcal{C}$. □

**Corollary I.0.4.** Let $d \in \mathbb{N}$ be arbitrary and let $\phi$ be a vector containing a permutation over $\{0, \dots, d\}$. Then

$$\max_{\phi} \mathcal{C}(\phi) = \lceil \frac{d^2}{2} \rceil + d$$

*Proof.* Let $\phi$ be the permutation with $\phi(i) = d - i$. Observe that $\phi$ is specially-ordered. The following holds with Corollary I.0.3.

$$\max_{\phi} \mathcal{C}(\phi) = \mathcal{C}(\phi) = \sum_{i=0}^{d} |i - (d-i)| = \sum_{i=0}^{d} |2i - d| = *$$

**Case 1: $d$ is even**

$$\begin{aligned} * &= \sum_{i=0}^{\frac{d}{2}} (d - 2i) + \sum_{i=\frac{d}{2}+1}^{d} (2i - d) \\ &= (\frac{d}{2} + 1)d - \sum_{i=0}^{\frac{d}{2}} 2i + \sum_{i=\frac{d}{2}+1}^{d} 2i - \frac{d}{2}d \\ &= d - 2\sum_{i=0}^{\frac{d}{2}} i + 2 \sum_{i=\frac{d}{2}+1}^{d} i \\ &= d - 2\frac{(\frac{d}{2}+1)\frac{d}{2}}{2} + 2\frac{(d+1)d}{2} - 2\frac{(\frac{d}{2}+1)\frac{d}{2}}{2} \\ &= d - 2(\frac{d}{2}+1)\frac{d}{2} + (d+1)d \\ &= \frac{d^2}{2} + d = \lceil \frac{d^2}{2} \rceil + d \end{aligned} \tag{6}$$

**Case 2: $d$ is odd**

$$\begin{aligned} * &= \sum_{i=0}^{\frac{d-1}{2}} (d - 2i) + \sum_{i=\frac{d+1}{2}}^{d} (2i - d) \\ &= \frac{d+1}{2}d - 2\sum_{i=0}^{\frac{d-1}{2}} i + 2 \sum_{i=\frac{d+1}{2}}^{d} i - \frac{d+1}{2}d \\ &= -2\frac{\frac{d-1}{2}(\frac{d-1}{2}+1)}{2} + 2\frac{d(d+1)}{2} - 2\frac{(\frac{d+1}{2}-1)\frac{d+1}{2}}{2} \\ &= \frac{d^2}{2} + d + \frac{1}{2} = \lceil \frac{d^2}{2} \rceil + d \end{aligned} \tag{7}$$

□

**Corollary I.0.5.**

$$\Delta(\text{votes}_1, \dots, \text{votes}_M) = \lceil \frac{M^2}{2} \rceil + M$$

*Proof.* This statement follows directly from Corollary I.0.4 for $d = M$. □

### C. Proof that DP Rank Aggregation is DP

**Lemma IV.2.** Algorithm 4 is DP with privacy budget $\varepsilon$.

*Proof.* This design leverages Subsample and Aggregate: We split the explanation data set $D$ into $l$ disjoint subsets (Line 3 of Algorithm 4) and aggregate the $l$ results with the borda voting rule (Line 8). Here, each ranking of the $M$ items based on a subset $\hat{D}_i$ is considered one "voter" who contributes to the end result. The items are ranked according to this vote in Line 11.

If the explanation data set changes in one record between $D$ and $D'$, then exactly one subset $\hat{D}_i$ and subsequently one voter is affected. The sensitivity of the borda voting rule to such a change is $\lceil \frac{M^2}{2} \rceil + M$ according to Corollary I.0.5. Thus, DP with privacy budget $\varepsilon$ is ensured by adding noise to the final vote count of each item in Line 9 with scale $\sigma = \frac{\lceil M^2/2 \rceil + M}{\varepsilon}$. □

### D. Proof of DP Enabler Theorem

We put forward a more formal definition of data flow diagrams which we will use in our proof:

**Definition I.2.** $G = (V, E)$ is called the data flow diagram for an algorithm $A$ with input $D$ if and only if

1) $G$ is a directed acylic graph (DAG).
2) $\exists! a \in V : E^-(v) = \emptyset$. Vertex $a$ is called the input of $A$ and is the only vertex with no incoming edges.
3) $\exists! b \in V : E^+(v) = \emptyset$. Vertex $b$ is called the output of $A$ and is the only vertex with no outgoing edges.
4) $\forall v \in V \ \exists O_v$, a domain.
5) $\forall v \in V \ \exists f_v : O_{w_1} \times \dots \times O_{w_m} \to O_v \in \mathit{Stages}(A)$ with $E^-(v) = \{w_1, \dots, w_m\}$ s.t. $A(D) = f_b \circ \dots \circ f_a$. In case a vertex $v \in V$ has no incoming edges, the arity of $f_v$ is 0, e.g., $f_a : O_a$.

Now, we will prove the main theorem.

**Theorem V.1.** Let $G = (V, E)$ be a data flow diagram for an algorithm $A$ with input $a \in V$ and output $b \in V$. Let $S \subseteq V$ be a set of vertices. Then $\forall A \in \mathcal{A}_G : S$ is a privacy-enabling set for $A$. $\Leftrightarrow S$ is an $s, t$ vertex separator of $G'$ with $G' = (\{s, t\} \cup V, \{(s, a), (b, t)\} \cup E)$.

*Proof.* We will first prove the implication from left to right by contraposition: Let $S$ be a privacy-enabling set for $G$ for any $A \in \mathcal{A}_G$. Assume that $S$ is not an $s, t$ vertex separator of $G'$. That means $s$ and $t$ are in the same connected component in $G' - S$.

**Case 1: There exists a path from $s$ to $t$ in $G' - S$.** Thus, there is a path from $a$ to $b$ in $G - S$ because $s$ is only connected to $a$ and $t$ only to $b$. Let that path be $(a, p_1, \dots, p_m, b)$. Consider an algorithm $A \in \mathcal{A}_G$ whose stages and final output only depend on this path. I.e., $f_b$ only depends on $f_{p_m}$ and any other inputs to $f_b$ do not change the outcome of $f_b$. The same applies to $f_{p_m}$ with respect to $f_{p_{m-1}}$ and so on. Additionally, $A$ shall not be differentially private, i.e., $\exists D, D', o \in O_b$ s.t. $\Pr[A(D) = o] > 0$ and $\Pr[A(D') = o] = 0$. Such

an algorithm $A$ exists, e.g., an algorithm which only computes the identity function for each stage along path $(a, p_1, \ldots, p_m, b)$. In that case, $o = D$. $A$ only depends on stages from $(a, p_1, \ldots, p_m, b)$ and none of these stages are in $S \Rightarrow \hat{A}_S = A$. $A$ is not differentially private $\Rightarrow \hat{A}_S$ is not differentially private. This is a contradiction to our assumption that $S$ is a privacy-enabling set.

**Case 2: There exists a path from $t$ to $s$ in $G' - S$.** Per definition, $t$ does not have any outgoing edges. Therefore, there cannot be a path from $t$ to $s$.

Now, we will prove the implication from right to left. Let $S$ be an $s, t$ vertex separator of $G'$. Consider an arbitrary but fixed algorithm $A \in \mathcal{A}_G$. There may be multiple ways to split $A$ into functions $f_v$ so that $G$ is a data flow diagram of $A$. Again, consider any one fixed configuration of functions $f_v$ s.t. $G$ is a data flow diagram of $A$.

Consider $\hat{A}_S$. We will now mark every vertex $v \in V$ for which we know that the associated function $f_v$ is differentially private in $\hat{A}_S$.

1) Mark every vertex $v \in S$. The functions $f_v$ are replaced with differentially private $\mathcal{M}(f_v)$ by definition of $\hat{A}_S$.
2) Per the post-processing theorem [2], we know that a function that only depends on differentially private inputs is differentially private itself. Therefore, mark all vertices for which all incoming edges come from marked vertices (unless the vertex represents the private input itself, i.e., vertex $a$): $\forall v \in V : v \neq a$ and $\forall u \in E^-(v) : u$ is marked $\Rightarrow$ mark $v$.
3) Repeat step 2 until the set of marked vertices stays constant.

This algorithm terminates as the number of marked vertices is strictly monotonically increasing until the end condition is met. There are now two possible scenarios:

**Case 1: $b$ is marked** This means that we know that the output of $f_b$ is differentially private and therefore $\hat{A}_S$ is differentially private.

**Case 2: $b$ is not marked** In this case, $b$ could still be a non-private result. However, we can bring this case to a contradiction. For any unmarked vertex $v \in V$, one of two conditions must be true. Either $v = a$ or $\exists (u, v) \in E$ and $u$ is not marked. Otherwise, $v$ would have been marked in step 2. We know that $b \neq a$, so the second condition is true. We can expand a (backwards) path from $b$ along unmarked vertices. In case there are multiple backwards-edges that lead to an unmarked vertex, choose one arbitrarily. The path must terminate at some point because $G$ and $G'$ are DAG. The last expanded vertex $w$ on the path has no backwards-edge to an unmarked vertex (otherwise we could continue expanding the path). Recalling the two possible conditions of unmarked vertices, we know that $w = a$. Thus, there is a path from $a$ to $b$ and therefore also from $s$ to $t$. This path contains no vertices from $S$ (otherwise, this vertex would have been marked). However, this is a contradiction to our assumption that $S$ is an $s, t$ vertex separator.

□

### E. Proof of Sensitivity for DP ALE

**Lemma V.2.**

$$\Delta\text{dif} = 2 \cdot (max_f - min_f). \tag{8}$$

*Proof.* There are two different cases to consider:

**Case 1** $\vec{x}^{(k)} \in D$ and $\vec{x}'^{(k)} \in D'$ fall into the same interval $i$, i.e., $\vec{x}^{(k)} \in X_i$ and $\vec{x}'^{(k)} \in X'_i$.

$$\begin{aligned}
\Delta\text{dif}_i &= \max_{D,D'} \|\text{dif}_i - \text{dif}'_i\|_1 \\
&= \max_{D,D'} \| \sum_{\vec{x} \in X_i} (f(\vec{x}_{\setminus j \leftarrow \hat{x}_i}) - f(\vec{x}_{\setminus j \leftarrow \hat{x}_{i-1}})) \\
&\quad - \sum_{\vec{x}' \in X'_i} (f(\vec{x}'_{\setminus j \leftarrow \hat{x}_i}) - f(\vec{x}'_{\setminus j \leftarrow \hat{x}_{i-1}}))\|_1 \\
&= \max_{D,D'} \|f(\vec{x}^{(k)}_{\setminus j \leftarrow \hat{x}_i}) - f(\vec{x}^{(k)}_{\setminus j \leftarrow \hat{x}_{i-1}}) \\
&\quad - f(\vec{x}'^{(k)}_{\setminus j \leftarrow \hat{x}_i}) + f(\vec{x}'^{(k)}_{\setminus j \leftarrow \hat{x}_{i-1}})\|_1 \\
&= \|max_f - min_f - min_f + max_f\|_1 \\
&= 2 \cdot (max_f - min_f)
\end{aligned} \tag{9}$$

The other intervals are unaffected and therefore have a sensitivity of 0. Thus, the total sensitivity of the dif vector is $2 \cdot (max_f - min_f)$.

**Case 2** $\vec{x}^{(k)} \in D$ and $\vec{x}'^{(k)} \in D'$ fall into two different intervals $\vec{x}^{(k)} \in X_i$ and $\vec{x}'^{(k)} X'_l$. First, consider the sensitivity of $\text{dif}_i$:

$$\begin{aligned}
\Delta\text{dif}_i &= \max_{D,D'} \|\text{dif}_i - \text{dif}'_i\|_1 \\
&= \max_{D,D'} \| \sum_{\vec{x} \in X_i} (f(\vec{x}_{\setminus j \leftarrow \hat{x}_i}) - f(\vec{x}_{\setminus j \leftarrow \hat{x}_{i-1}})) \\
&\quad - \sum_{\vec{x}' \in X'_i} (f(\vec{x}'_{\setminus j \leftarrow \hat{x}_i}) - f(\vec{x}'_{\setminus j \leftarrow \hat{x}_{i-1}}))\|_1 \\
&= \max_{D,D'} \|f(\vec{x}^{(k)}_{\setminus j \leftarrow \hat{x}_i}) - f(\vec{x}^{(k)}_{\setminus j \leftarrow \hat{x}_{i-1}})\|_1 \\
&= \|max_f - min_f\|_1 \\
&= max_f - min_f
\end{aligned} \tag{10}$$

The sensitivity analysis of $\text{dif}_l$ can be done analogously and also results in sensitivity $max_f - min_f$. Other intervals are unaffected and have a sensitivity of 0. Thus, the total sensitivity of dif also comes out as $2 \cdot (max_f - min_f)$.

□

### F. Proof of Sensitivity for DP PFI

**Lemma V.3.**

$$\Delta\text{error}_j = \frac{2 \cdot (max_f - min_f)^2}{n} \tag{11}$$

*Proof.*

$$
\begin{aligned}
\Delta\text{error}_j &= \max_{D,y,D',y'} \|L(f,\hat{D},y) - L(f,\hat{D}',y')\|_1 \\
&= \max_{D,y,D',y'} \|\frac{1}{n}\sum_{i=1}^{n}\left(y_i - f(\vec{x}^{(i)}_{\setminus j \leftarrow \vec{x}_j^{(\phi(i))}})\right)^2 \\
&\quad - \frac{1}{n}\sum_{i=1}^{n}\left(y'_i - f(\vec{x}'^{(i)}_{\setminus j \leftarrow \vec{x}'^{(\phi(i))}_j})\right)^2 \|_1 \\
&= \max_{D,y,D',y'} \|\frac{1}{n}((y_k - f(\vec{x}^{(k)}_{\setminus j \leftarrow \vec{x}_j^{(\phi(k))}}))^2 \\
&\quad - (y'_k - f(\vec{x}'^{(k)}_{\setminus j \leftarrow \vec{x}'^{(\phi(k))}_j}))^2 \\
&\quad + (y_{\phi^{-1}(k)} - f(\vec{x}^{(\phi^{-1}(k))}_{\setminus j \leftarrow \vec{x}_j^{(k)}}))^2 \\
&\quad - (y'_{\phi^{-1}(k)} - f(\vec{x}'^{(\phi^{-1}(k))}_{\setminus j \leftarrow \vec{x}'^{(k)}_j}))^2 \|_1 \\
&\leq \|\frac{1}{n}(2\cdot(max_f - min_f)^2 + 2\cdot 0)\|_1 \\
&= \frac{2\cdot(max_f - min_f)^2}{n}
\end{aligned}
\tag{12}
$$

□

## II. Further Explainers

This section covers the application of the DP enabler theorem to Partial Dependence Plots and it verifies that Generic DP Plots can be applied to SHAP Dependence Plot and Generic DP Rank Aggregation can be applied to Feature Interaction.

### A. Differentially Private Partial Dependence Plots

We apply the DP enabler theorem to a third explainer not covered in the main body of the paper. First, we introduce the explainer. Then, we find all minimal privacy-enabling sets. Finally, we argumentatively identify the most promising option from the possible detected explainer-specific design and implement it.

*1) Partial Dependence Plot:* A Partial Dependence Plot (PDP) [1] depicts how a single feature influences the predictions of the model. Algorithm 7 provides the pseudo code of PDP. PDPs rely on the partial dependence function.

**Definition II.1.** The partial dependence function $PD_j$ for a feature with index $j$ has three inputs: a feature value $x$, the prediction function $f$ of the model, and the explanation data set $D$, as follows:

$$PD_j(x,f,D) := \frac{1}{n}\sum_{\vec{x}\in D} f(\vec{x}_{\setminus j \leftarrow x}) \tag{13}$$

Intuitively, it estimates the average prediction of $f$ for samples with feature value $x$ for feature $j$.

The input of Algorithm 7 is the prediction function $f$ of the model one wishes to explain, the feature index $j$ whose partial dependence will be displayed, and the explanation data set $D$. For PDP, *x values* and *y values* refer to points on the plotted output which are defined by their position on the x and y axis. The algorithm determines all unique values of feature $j$ in $D$ (Line 2). These are the x values which are included in the final plot. For each x value, the corresponding y value (i.e., the corresponding partial dependence) is calculated in Lines 4 to 7. Finally, the algorithm outputs a set of x and y values. These points are visualized as a plot. In case of a categorical feature, the plot is a bar chart with one bar for each category. The PDP should include a rug plot that shows the empirical distribution of feature $j$ in the explanation data set. This helps the user gauge in which regions the plot accurately describes the behavior of the model [3].

**Algorithm 7** Partial Dependence Plot

**Input:** Prediction function $f$ of the ML model to be explained, feature index $j$, explanation data set $D$

1: **procedure** PartialDependencePlot($f, j, D$)
2: $\quad\hat{x} \leftarrow \{x_j | \vec{x} \in D\}$
3: $\quad$output $\leftarrow \emptyset$
4: $\quad$**for** $x \in \hat{x}$ **do**
5: $\quad\quad y \leftarrow PD_j(x, f, D)$
6: $\quad\quad$output $\leftarrow (x, y) \cup$ output
7: $\quad$**end for**
8: $\quad$**return** output
9: **end procedure**

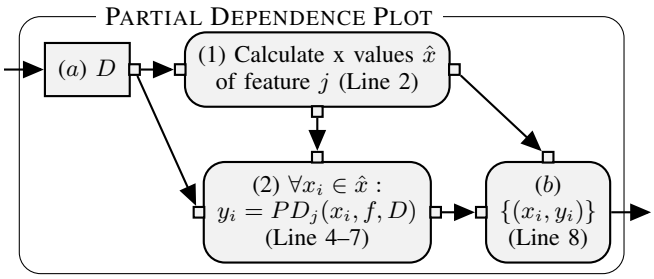



Fig. 1: Data flow diagram of the Partial Dependence Plot. Line numbers indicate corresponding lines in Algorithm 7.

*2) Detection:* The data flow diagram in Figure 1 depicts how the stages of PDP (see Algorithm 7) depend on the explanation data set $D$. The minimal privacy-enabling sets are $\{a\}$, $\{b\}$ and $\{(1),(2)\}$.

From among these options, we reject two, for the following reasons. *X values* and *y values* refer to points on the plotted output:

- Adding noise to the output directly is undesirable: The sensitivity of the output is dominated by the maximum sensitivity of the x or the y values. Thus, if one perturbs the output, this may add more noise to either the x or the y values than necessary. This issue does not occur if one perturbs the x and y values separately.
- Adding noise to the input requires perturbing every single sample in $D$ (or generating a whole new data set). This alters every input for the prediction function $f$. It is used by the explainer $|\hat{x}| \cdot n$ times. In contrast, our sensitivity analysis will show that the required noise for the third option only grows proportionally to $|\hat{x}| = m$.

Thus, we perturb the calculation of the x values (Stage (1)) and the calculation of the y values (Stage (2)).

*3) Implementation:* Algorithm 8 is DP PDP. It takes as input the prediction function $f$ of the model and the index $j$ of the feature to be explained, the explanation data set $D$, the resolution of the final plot $m$, the privacy budget $\varepsilon$ and the domain of feature $j$ as a set of categories $C_j$ for a categorical feature or numeric bounds for a continuous feature. The algorithm makes use of the partial dependence function. We now explain this algorithm and how it guarantees DP.

**Algorithm 8** DP PDP

**Input:** Prediction function $f$, feature index $j$, explanation data set $D$, resolution $m$, privacy budget $\varepsilon$, categories $C_j$, bounds $min_j, max_j, min_f, max_f$

1: **procedure** DPPARTIALDEPENDENCE-PLOT($f, j, D, m, \varepsilon, C_j, min_j, max_j, min_f, max_f$)
2: **if** feature $j$ is categorical **then**
3: $\hat{x} \leftarrow C_j$
4: **else if** feature $j$ is continuous **then**
5: $\hat{x} \leftarrow \{min_j + \frac{k}{m-1}(max_j - min_j) \mid k \in \{0, \dots, m-1\}\}$
6: **end if**
7: output $\leftarrow \emptyset$
8: $\sigma_y \leftarrow \frac{m \cdot (max_f - min_f)}{n \cdot \epsilon}$
9: **for** $x \in \hat{x}$ **do**
10: $y \leftarrow PD_j(x, f, D) + \text{Lap}(\sigma_y)$
11: output $\leftarrow (x, y) \cup$ output
12: **end for**
13: **return** output
14: **end procedure**

*a) X Values for DP PDP:* The x values for a continuous feature are equidistant points between $min_j$ and $max_j$, see Line 5. Equidistant x values have the advantage that they do not need any privacy budget, as they are independent of the data set. As an alternative, we have considered to choose the x values based on a DP Kernel Density Estimation. However, preliminary experiments have revealed this estimation to require a large part of the privacy budget while not increasing the utility of the plot.

For a categorical feature, the given categories $C_j$ are used as x values. This does not require any privacy budget either.

*b) Y Values for DP PDP:* The y values of DP PDP can be calculated in the same way as for the original PDP explanation (Lines 4 to 7 of Algorithm 7). We then deploy the Laplace Mechanism to make the y values differentially private. We have derived the following upper bound for the sensitivity of calculating one y value in a PDP at a specific value $x$ for feature $j$.

**Lemma II.1.**

$$\Delta(PD_i(x_1, f, D), \dots, PD_i(x_m, f, D)) \leq \frac{m}{n}(max_f - min_f) \tag{14}$$

*Proof.*

$$\Delta PD_j = \max_{D,D'} \|PD_j(x, f, D) - PD_j(x, f, D')\|_1 \tag{15}$$

$$= \max_{D,D'} \|\frac{1}{n} \sum_{\vec{x} \in D} f(\vec{x}_{\setminus j \leftarrow x}) - \frac{1}{n} \sum_{\vec{x}' \in D'} f(\vec{x}'_{\setminus j \leftarrow x})\|_1 \tag{16}$$

$$= \max_{D,D'} \|\frac{1}{n} \left( f(\vec{x}^{(k)}_{\setminus j \leftarrow x}) - f(\vec{x}'^{(k)}_{\setminus j \leftarrow x}) \right) \|_1 \tag{17}$$

$$\leq \|\frac{1}{n}(max_f - min_f)\|_1 \tag{18}$$

Then the total sensitivity if one calculates the y values for all $m$ selected x values is

$$\Delta(PD_i(x_1, f, D), \dots, PD_i(x_m, f, D)) \leq \frac{m}{n}(max_f - min_f) \tag{19}$$

□

Thus, we can make the y values differentially private by adding Laplacian noise with scale $m \cdot (max_f - min_f)/(n \cdot \varepsilon)$ (see Line 10).

We can visualize the final output of points in the same way as for the non-private PDP.

*c) Differentially Private Histogram:* Same as ALE, the rug plot of the original PDP must be altered as it also depends on the explanation data set and can leak private data. We use a DP histogram query [2] analogously to Section V-B2c in the main paper.

### B. Application of Generic DP Plots to SHAP Dependence Plot

Here, we verify that DP Generic Plots can be applied to the SHAP Dependence Plot.

SHAP [4] is a local explainer that gives an importance to each feature of a record $\vec{x}$ for the prediction $f(\vec{x})$ of the model $f$. One can also derive a global explanation from SHAP: The SHAP Dependence Plot [5] visualizes the SHAP value of each record $\vec{x} \in D$ for a specific feature $j$ in a scatter plot. The x value of a point in the scatter plot refers to the records feature value $\vec{x}_j$, while the y value is the SHAP value.

SHAP Dependence Plot can be implemented in such a way that the 2D-points of the scatter plot are returned as the output of the explainer. This output matches the output required by DP Generic Plots. As input, the SHAP Dependence Plot requires the model $f$ to explain, the explanation data set $D$ and the feature index $j$. The definition of a plot explainer requires the input to only consist of $f$ and $D$. However, similarly to PDP and ALE, we can parameterize the SHAP Dependence Plot with feature $j$ before handing it over to DP Generic Plots. Implementation-wise, this can be realized with a simple interface that only requires $f$ and $D$ as input and then calls SHAP Dependence Plot with the feature index you wish to explain.

The plot produced by Generic DP Plots would approximate an average of the original scatter plot. The private plot would not contain information about the variation in SHAP value between different records with the same feature value $\vec{x}_j$. Here, an explainer-specific design could provide a more accurate

approximation. However, the generic design will capture the general trend of the SHAP Dependence Plot as a private line plot. Thus, Generic DP Plots can be applied to SHAP Dependence Plot.

### C. Application of Generic DP Ranking to Feature Interaction

In this section, we verify that Generic DP Rank Aggregation can be applied to Feature Interaction.

Features interact when the prediction of a model $f$ cannot be decomposed additively into functions, each only depending on a single feature. Rather, the influence of a single feature on the prediction also depends on the values of other features [3]. A common way to measure the interaction between two or more features is Friedman's H-statistic [6].

A feature interaction explainer can (similarly to PFI) provide a ranking of features. This ranking is based on how strongly each feature interacts with one chosen feature or based on how strongly each feature interacts with all other features. This is particularly relevant as interpreting the H-statistic for a feature by itself (without relation to other features) can be difficult [3]. In this case, the explainer returns a ranking of features as output, just as required by the definition of a rank explainer. The input also adheres to the definition. The feature interaction explainer requires model $f$ and explanation data set $D$ when calculating the interaction with every other feature. If the interaction is measured with respect to one specific feature, the feature interaction explainer additionally requires the index $j$ of that feature. Once again, the explainer may be parameterized with that information in order to adhere to the definition of a rank explainer (analogous to Section II-B). Thus, Generic DP Rank Aggregation can be applied to Feature Interaction.

## III. Results

This section includes new results for DP PDP, in addition to all results of Experiment 1 and 2 that were not covered in the paper. These results contain evaluations for an additional privacy budget of $\varepsilon = 10$. This has been left out of the paper as such high privacy budgets are believed to offer hardly any privacy at all [7]. However, we include these additional results for completeness here.

### A. Results of Experiment 1 for PDP

These results are completely new as DP PDP has not been discussed in the main body of the paper. Similarly to ALE, we refer to the instantiation of Generic DP Plots with PDP as Generic DP PDP.

Table I shows which private explainer has a lower MISE for each examined feature and privacy budget $\varepsilon$. DP PDP has a lower MISE than Generic DP PDP for all evaluated values of $\varepsilon$ for 36 of the 38 features of the three data sets. Clearly, DP PDP achieves a better privacy-utility trade-off than Generic DP PDP in almost all cases. We will examine the reasons for the two deviating features below.

Figure 2 shows the MISE results for each most important feature of the three data sets. The blue and red lines plot the results for the generic and explainer-specific designs, respectively.

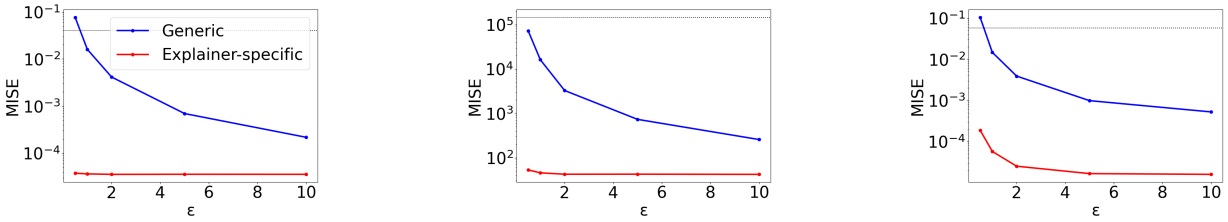
| Dataset | Feature | DP PDP | Generic DP PDP |
|---|---|---|---|
| Census Income | 11 Features | $\varepsilon \in \{0.5, 1, 2, 5, 10\}$ | $\varepsilon \in \emptyset$ |
| | Capital Gain | $\varepsilon \in \{0.5, 1, 2, 5\}$ | $\varepsilon \in \{10\}$ |
| | Education Years | $\varepsilon \in \{0.5, 1, 2, 5\}$ | $\varepsilon \in \{10\}$ |
| Bike Sharing | All 10 Features | $\varepsilon \in \{0.5, 1, 2, 5, 10\}$ | $\varepsilon \in \emptyset$ |
| Heart Disease | All 15 Features | $\varepsilon \in \{0.5, 1, 2, 5, 10\}$ | $\varepsilon \in \emptyset$ |

TABLE I: Which explainer has lower MISE for the respective feature and privacy budget $\varepsilon$.

Figure 3 contains plots for the feature "Hour" of the Bike Sharing data set compared to the original explanation in grey. These plots exemplify the effect of increasing the privacy parameter $\varepsilon$. DP PDP yields an accurate representation for all examined values of $\varepsilon$. Here, DP PDP provides good utility for comparatively small values of $\varepsilon$, therefore providing good privacy. In contrast, the explanation from Generic DP PDP shows larger discrepancies from the original plot. This is most apparent for $\varepsilon = 0.5$. Generic DP PDP offers utility only for moderate levels of $\varepsilon$ for this feature. However, unlike the generic instantiation, DP PDP shows little to no improvement when increasing the privacy budget $\varepsilon$. The private plot deviates from the original plot for both peaks of the predicted number of bikes. This is because a resolution of 20 for the private plot cannot capture all 24 hours on the x-axis of the original plot. As we will see in the second experiment in Section III-B, the results further improve for DP PDP for resolution $m = 50$.

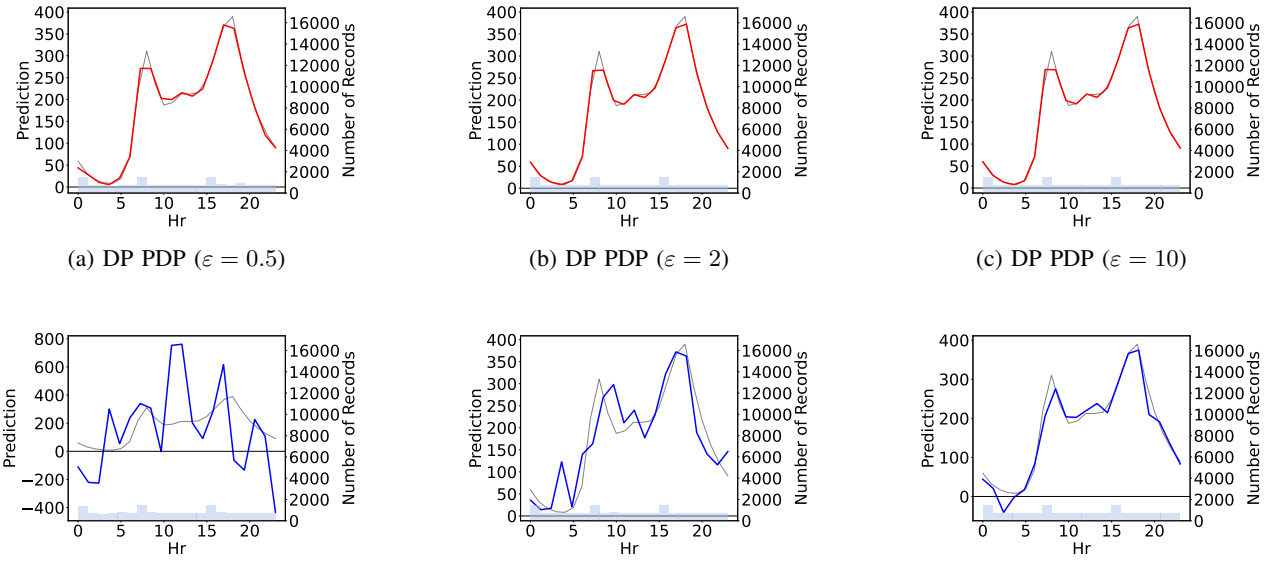
There are two features for which Generic DP PDP partially outperforms DP PDP: Generic DP PDP has a lower MISE than DP PDP for $\varepsilon = 10$ for "Capital Gain" and "Education Years". In both cases, we observe that the use of equidistant x values is not optimal and that interpolation by Generic DP PDP can smooth the final plot. This interpolation can be helpful in the face of a low resolution such as $m = 20$. The explanation for feature Capital Gain illustrates this effect in Figure 4. The line markers show the x and y-points that are given as output by DP PDP and Generic DP PDP. First, one can observe that the use of equidistant x values is not optimal for this plot. The slope between 0$ and 20,000$ would require a higher resolution to be accurately depicted, while a lower resolution would suffice for the remaining plateau after 20,000$. Second, considering the slope between 0$ and 20,000$, the x, y-points of DP PDP match almost exactly with the original plot, compared to the x, y-points of Generic DP PDP in particular. This results in a larger mismatch *between* these x, y-points compared to Generic DP PDP, which interpolates between 200 plots. However, the improvement over DP PDP is relatively small, as Generic DP PDP has worse accuracy for the remaining plateau of the plot. In general, we would recommend using DP PDP, as its result improves when one increases the resolution $m$. We investigate this in Experiment 2 in Section III-B.

Figure 11e shows the result for feature Education Years.




(a) Census Income - Age (b) Bike Sharing - Hour (c) Heart Disease - Age

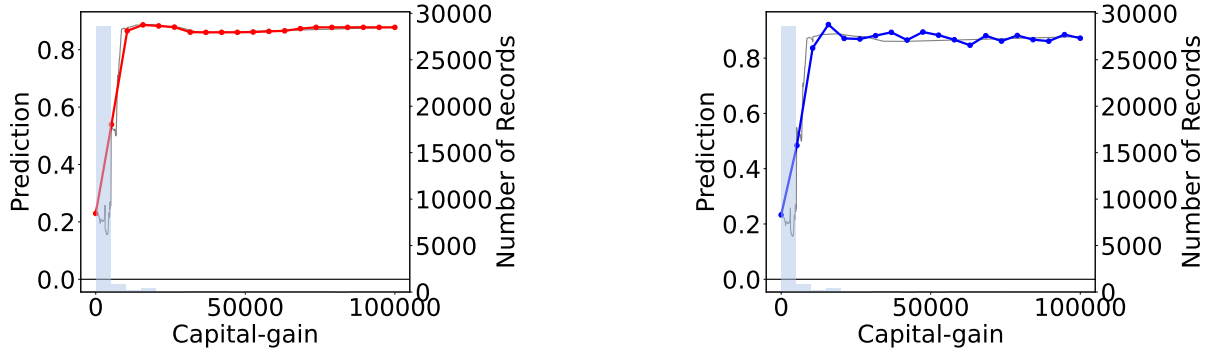
Fig. 2: Main results for PDP




(a) DP PDP ($\varepsilon = 0.5$) (b) DP PDP ($\varepsilon = 2$) (c) DP PDP ($\varepsilon = 10$)

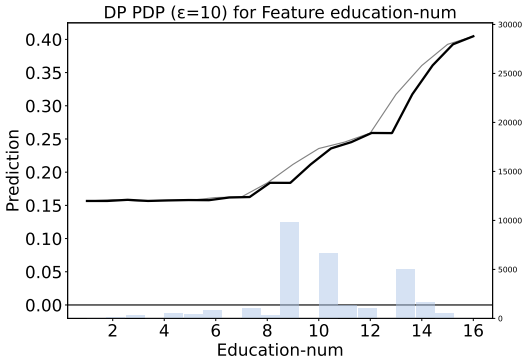
(d) Generic DP PDP ($\varepsilon = 0.5$) (e) Generic DP PDP ($\varepsilon = 2$) (f) Generic DP PDP ($\varepsilon = 10$)

Fig. 3: Qualitative Results for PDP

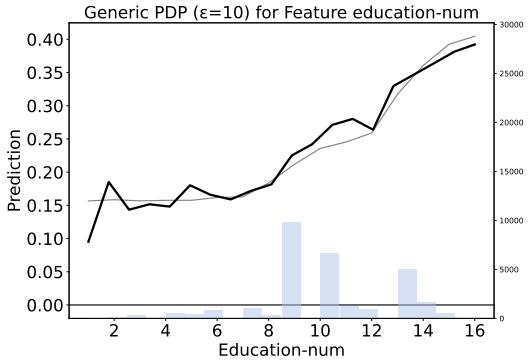



(a) DP PDP ($\varepsilon = 10$) (b) Generic DP PDP ($\varepsilon = 10$)

Fig. 4: Feature Capital Gain




(a) DP PDP ($\varepsilon = 10$)




(b) Generic DP PDP ($\varepsilon = 10$)

Fig. 5: Feature Education Years

Generic DP PDP has a lower MISE than DP PDP for $\varepsilon = 10$. The feature Education Years only consists of integer values. Using 20 equidistant x values however means that they are not integer values any more. This is unproblematic for Generic DP PDP as it simply interpolates between the nearest integer x values included in the plot of each split. DP PDP however enters these non-integer values into the model (here: a random forest). The splits of the trees in the random forest only consider integer values as this is the form of its training data. For instance, it does not make a difference whether the input is '1' or '1.25', all trees will make the same prediction. So the DP PDP plot appears to be "lagging behind" the original plot as some y values are repeated, see Figure 5. The original non-private PDP is shown in grey.

### B. Results of Experiment 2 for PDP

Figure 6 shows the results for each most important feature. Generic DP PDP tends to perform worse for higher resolutions. The result is more varied for DP PDP. Here, higher resolutions start out with a higher MISE while the privacy budget is low ($\varepsilon \in \{0.5, 1\}$). As the budget increases ($\varepsilon \in \{5, 10\}$), higher resolutions start to have a lower MISE. For feature "Age" in Census Income for instance, resolution 100 is worst for $\varepsilon = 0.5$ but best for $\varepsilon = 10$. Overall, higher resolutions resulted in a worse privacy-utility trade-off for Generic DP PDP. For DP PDP, the choice of resolution for these data sets depends on whether utility or privacy is valued more. If one values utility over privacy, these findings suggest choosing a higher resolution with less privacy. A lower resolution provides better utility for high levels of privacy.

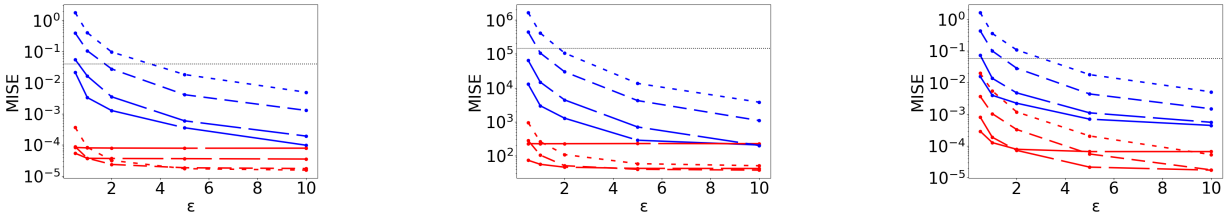
*a) Feature Capital Gain:* In addition to the most important features according to PFI, we have studied the effects of different resolutions for "Capital Gain". In Experiment 1, Generic DP PDP performed better than DP PDP for a resolution of $m = 20$. We expected this result to change for higher resolutions. In Figure 7, higher resolutions start with a comparatively high MISE and yield the lowest MISE for higher budgets $\varepsilon$, even for the generic designs. As expected, DP PDP performs poorly for low resolutions (i.e., 10, 20). Performance improves and beats the generic version for resolutions 50 and 100. This confirms our hypothesis that DP PDP can offer a better privacy-utility trade-off than the generic design for this feature if the resolution is increased.

A similar effect was observed for DP ALE in the paper. Figure 8 shows qualitative results for feature Capital Gain with higher resolutions. Both DP PDP and DP ALE can capture the beginning of the plot better for higher resolutions compared to smaller resolutions. The imporvement is a lot more pronounced for DP ALE.

### C. Detailed Discussion of Results of ALE

In the paper, Generic DP ALE partially outperforms DP ALE for two features: It has a lower MISE than DP ALE for $\varepsilon \in \{2, 5\}$ for "Capital Gain" of Census Income and for $\varepsilon \in \{0.5, 1\}$ for "Native Country" in the same data set. Considering also results for $\varepsilon = 10$ adds a third feature to this set of deviating results: Generic DP ALE outperforms DP ALE for $\varepsilon = 10$ for feature "Capital Loss" in the Census Income data set. We now discuss the reasons behind these results, in more detail than in the paper.

The deviating result for Capital Loss happens for the same reason as for Capital Gain as discussed in the paper: Generic DP ALE outperforms DP ALE because of the equidistant x values of the former explainer. This is due to feature distributions that are concentrated on a small part of the feature space. This is exemplified by the plots for Capital Gain in Figure 9. This feature has a very lopsided distribution in the training data with 92% of participants having a capital gain of \$0. So the vast majority of the 20 quantiles for DP ALE are at or close to 0, due to addition of uniform noise to the feature values in the algorithm. Most of the plot (between values of \$1 up to \$100 000) has a very low resolution for DP ALE. This is not the case for Generic DP ALE where x values are equidistant. Thus, Generic DP ALE better approximates the original baseline ALE that has a higher resolution of 100. However, the shortcomings of DP ALE can be fixed by using a higher resolution, even though this increases the random noise required. This was investigated in Experiment 2




(a) Census Income - Age (b) Bike Sharing - Hour (c) Heart Disease - Age

Fig. 6: Results of experiment 2 for PDP

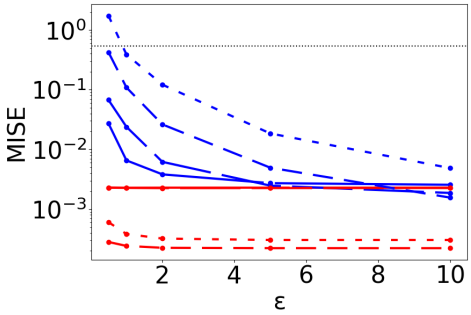



Fig. 7: Results of experiment 2 for feature "Capital Gain" for PDP

in Section VII-B of the main paper. More qualitative results for DP ALE for Capital Gain with higher resolutions can be found in Figure 8.

The diverging result for Native Country occurs because some categories (i.e., countries) have very few members in the explanation data set. With ALE, each effect only depends on records from one category. If a category has few members, a single record can have a larger impact on the final result. We see this for both private explanations: For DP ALE, Laplacian noise of same scale is added to each y value before it is divided by the number of records in the category. So this noise will have a larger relative impact for categories with few records than for categories with many records. For Generic DP ALE, y values for small categories are also estimated less accurately: The fewer members a category has, the more unstable its results are in the different subsets. This negatively affects the aggregated final result. Therefore, both designs perform poorly for the feature Native Country and are dominated by randomly added noise. See the qualitative results in Figure 10. This issue does not occur for continuous features because of the use of quantiles. There, each effect depends on an (approximately) equal number of records from the explanation data set. The issue also does not occur for PDP. Namely, PDP uses every record to estimate the partial dependence of a category, instead of just records from one category. Each y value in the plot always depends on all records from the data set. So categories with few members are not affected disproportionately by the added random noise for private PDPs. However, the use of all records to calculate the partial dependence in a PDP can result in misguiding explanations [8].

### D. All Results of Experiment 1 for Plot Explainers

Each result for each feature for the plot explainers can be seen in Figure 11, Figure 12, Figure 13, Figure 14, Figure 15 and Figure 16.

## References


[1] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.

[2] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy." *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3-4, pp. 211–407, 2014.

[3] C. Molnar, *Interpretable Machine Learning*, 2nd ed. Lulu.com, 2022. [Online]. Available: https://christophm.github.io/interpretable-ml-book

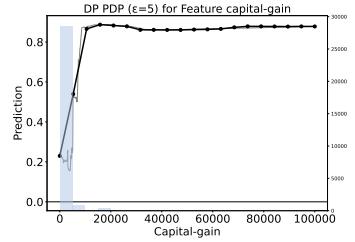
[4] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. NY, USA: Curran Associates, Inc., 2017, pp. 4765–4774. [Online]. Available: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

[5] S. M. Lundberg, G. G. Erion, and S.-I. Lee, "Consistent individualized feature attribution for tree ensembles," *arXiv preprint arXiv:1802.03888*, 2018.

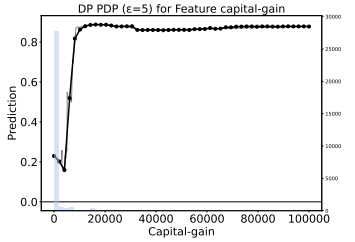
[6] J. H. Friedman and B. E. Popescu, "Predictive learning via rule ensembles," *The annals of applied statistics*, pp. 916–954, 2008.

[7] A. Greenberg, "How one of apple's key privacy safeguards falls short," Wired, Sep. 2017. [Online]. Available: https://www.wired.com/story/apple-differential-privacy-shortcomings/

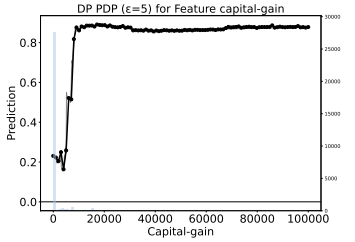
[8] D. W. Apley and J. Zhu, "Visualizing the effects of predictor variables in black box supervised learning models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 82, no. 4, pp. 1059–1086, jun 2020.



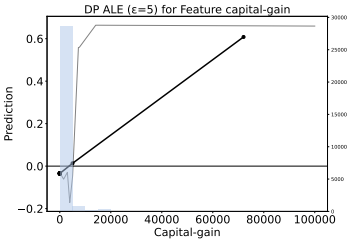

(a) DP PDP ($m = 20$, $\varepsilon = 5$)



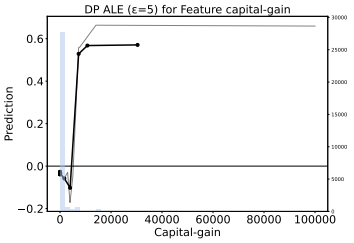

(b) DP PDP ($m = 50$, $\varepsilon = 5$)



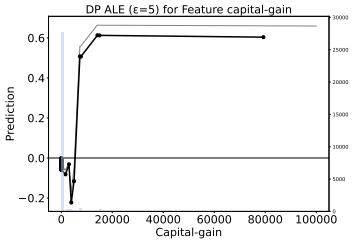

(c) DP PDP ($m = 100$, $\varepsilon = 5$)




(d) DP ALE ($m = 20$, $\varepsilon = 5$)

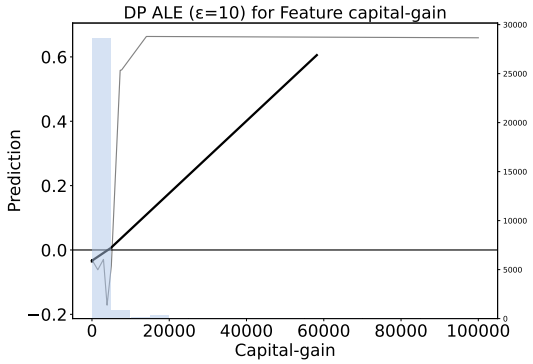



(e) DP ALE ($m = 50$, $\varepsilon = 5$)

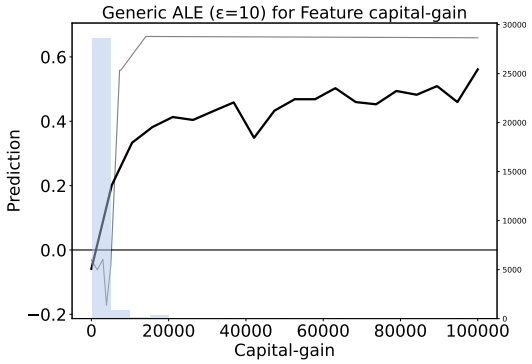



(f) DP ALE ($m = 100$, $\varepsilon = 5$)

Fig. 8: Qualitative results of experiment 2 for feature Capital Gain

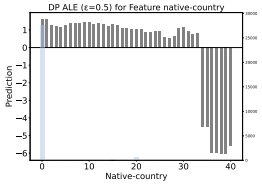



(a) DP ALE ($\varepsilon = 10$)

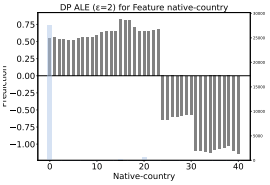



(b) Generic DP ALE ($\varepsilon = 10$)

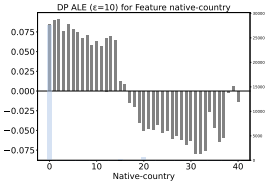
Fig. 9: Feature 'capital-gain'



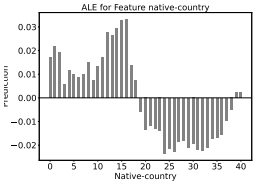

(a) DP ALE ($\varepsilon = 0.5$)



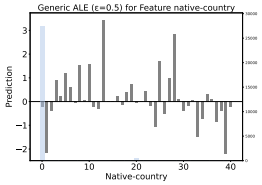

(b) DP ALE ($\varepsilon = 2$)



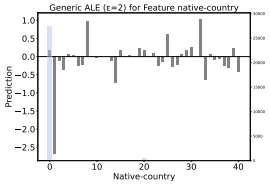

(c) DP ALE ($\varepsilon = 10$)



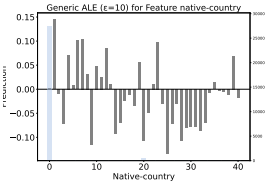

(d) Non-private baseline



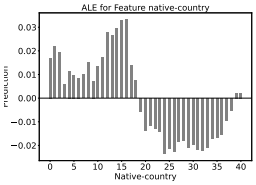

(e) Generic DP ALE ($\varepsilon = 0.5$)




(f) Generic DP ALE ($\varepsilon = 2$)

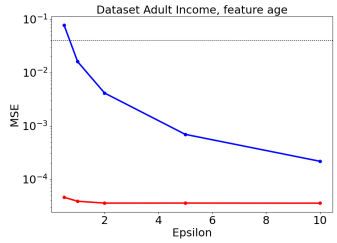



(g) Generic DP ALE ($\varepsilon = 10$)

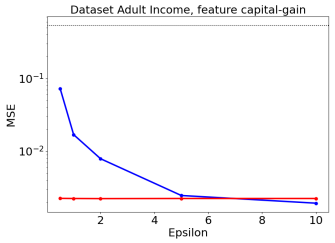



(h) Non-private baseline

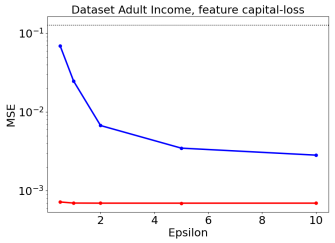
Fig. 10: Feature Native Country



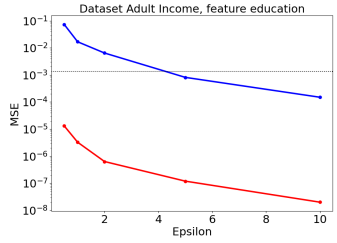

(a) PDP; Census Income; 'age'



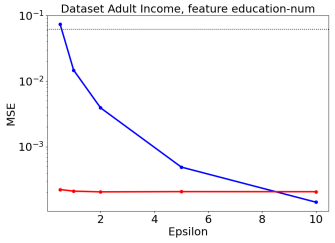

(b) PDP; Census Income; 'capital-gain'



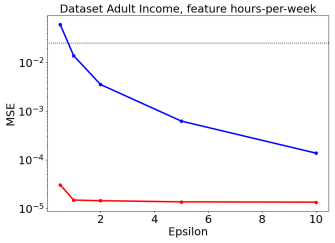

(c) PDP; Census Income; 'capital-loss'



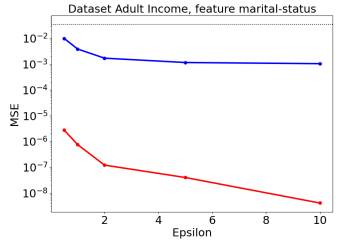

(d) PDP; Census Income; 'education'



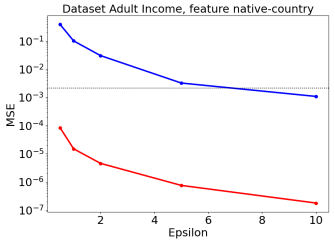

(e) PDP; Census Income; 'education-num'



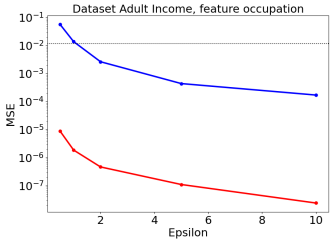

(f) PDP; Census Income; 'hours-per-week'



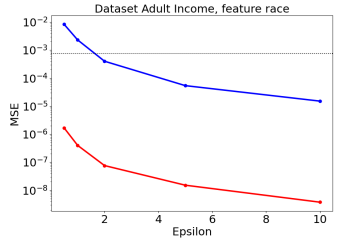

(g) PDP; Census Income; 'marital-status'



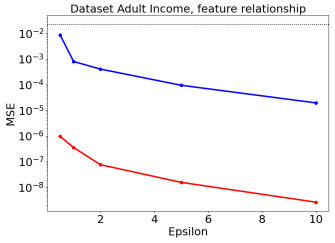

(h) PDP; Census Income; 'native-country'



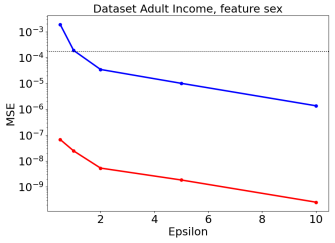

(i) PDP; Census Income; 'occupation'



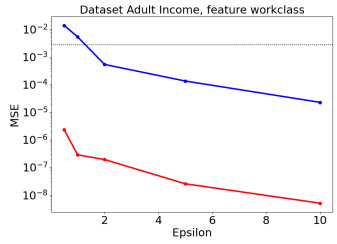

(j) PDP; Census Income; 'race'



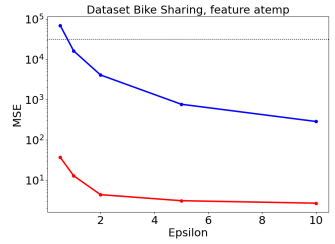

(k) PDP; Census Income; 'relationship'



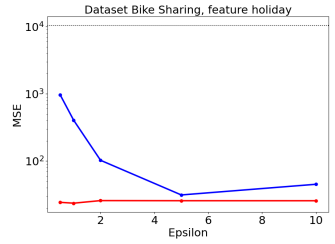

(l) PDP; Census Income; 'sex'




(m) PDP; Census Income; 'workclass'

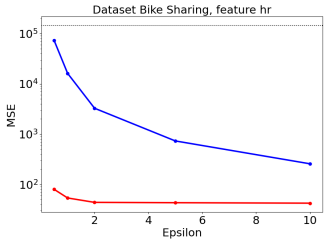



(n) PDP; Bike Sharing; 'atemp'

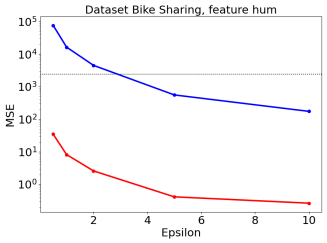



(o) PDP; Bike Sharing; 'holiday'

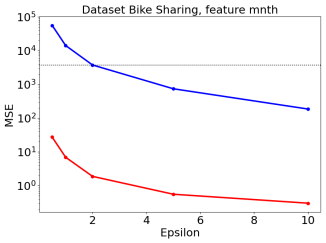
Fig. 11: All results of Experiment 1 for DP PDP and Generic DP PDP.



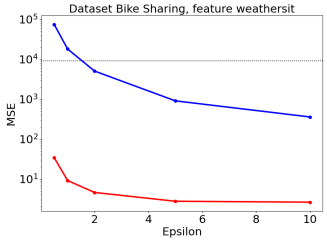

(a) PDP; Bike Sharing; 'hr'



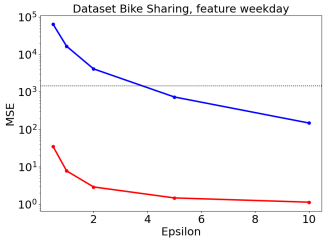

(b) PDP; Bike Sharing; 'hum'



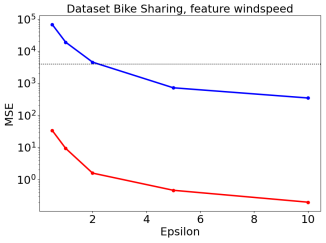

(c) PDP; Bike Sharing; 'mnth'



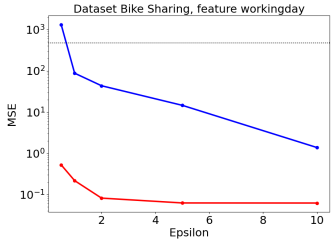

(d) PDP; Bike Sharing; 'weathersit'



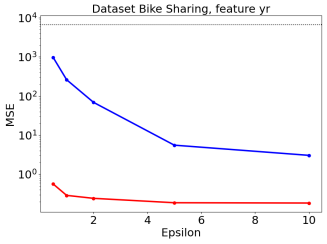

(e) PDP; Bike Sharing; 'weekday'



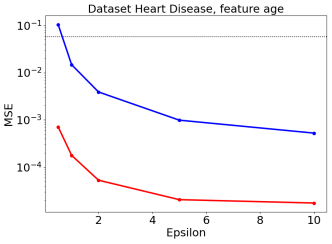

(f) PDP; Bike Sharing; 'windspeed'



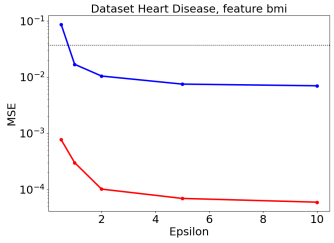

(g) PDP; Bike Sharing; 'workingday'



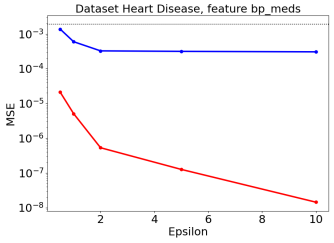

(h) PDP; Bike Sharing; 'yr'



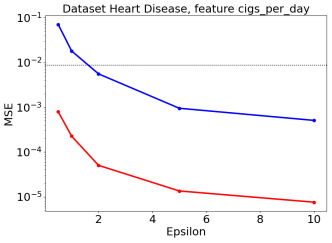

(i) PDP; Heart Disease; 'age'



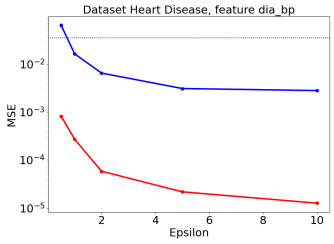

(j) PDP; Heart Disease; 'bmi'



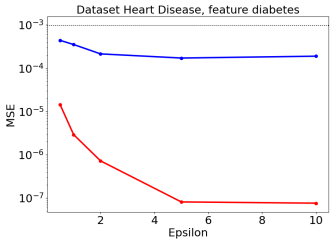

(k) PDP; Heart Disease; 'bp_meds'



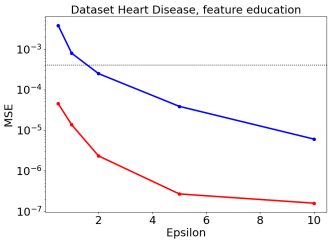

(l) PDP; Heart Disease; 'cigs_per_day'




(m) PDP; Heart Disease; 'dia_bp'

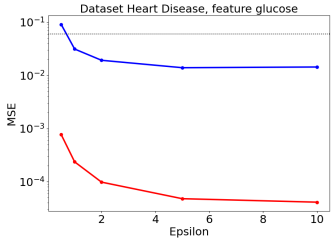



(n) PDP; Heart Disease; 'diabetes'

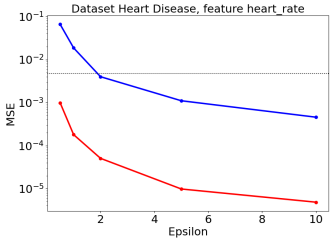



(o) PDP; Heart Disease; 'education'

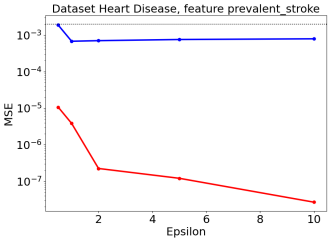
Fig. 12: All results of Experiment 1 for DP PDP and Generic DP PDP.



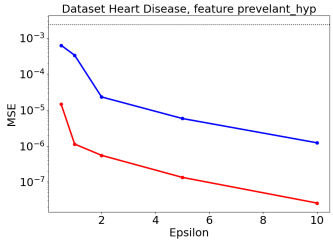

(a) PDP; Heart Disease; 'glucose'



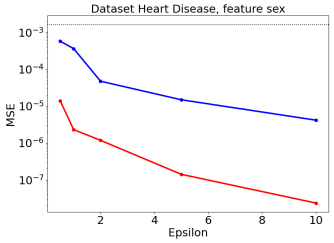

(b) PDP; Heart Disease; 'heart_rate'



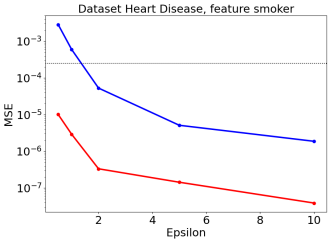

(c) PDP; Heart Disease; 'prevalent_stroke'



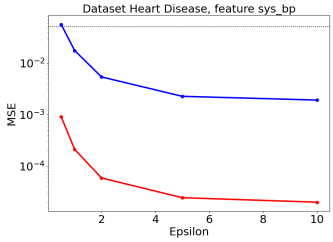

(d) PDP; Heart Disease; 'prevelant_hyp'



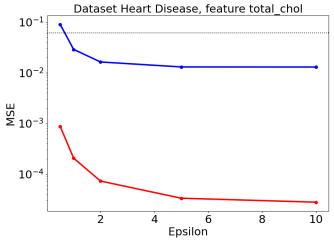

(e) PDP; Heart Disease; 'sex'




(f) PDP; Heart Disease; 'smoker'

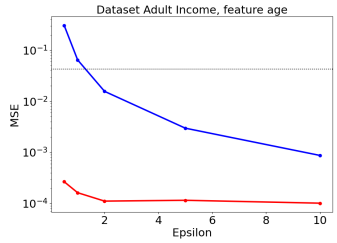



(g) PDP; Heart Disease; 'sys_bp'

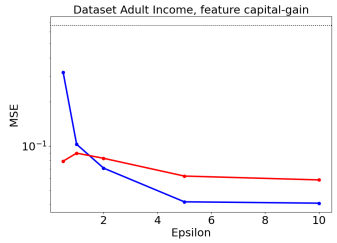



(h) PDP; Heart Disease; 'total_chol'

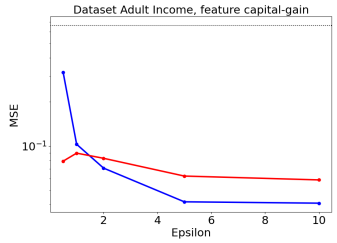
Fig. 13: All results of Experiment 1 for DP PDP and Generic DP PDP.



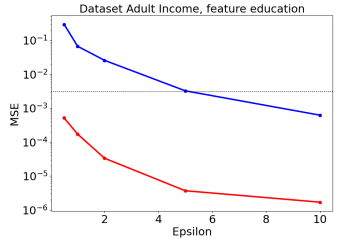

(a) ALE; Census Income; 'age'



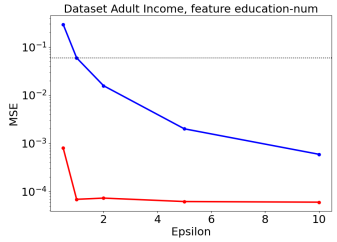

(b) ALE; Census Income; 'capital-gain'



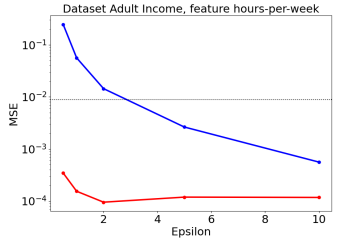

(c) ALE; Census Income; 'capital-loss'



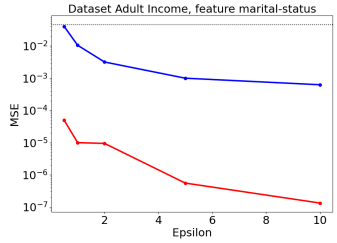

(d) ALE; Census Income; 'education'



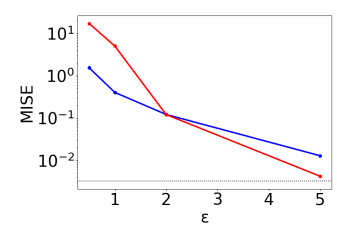

(e) ALE; Census Income; 'education-num'



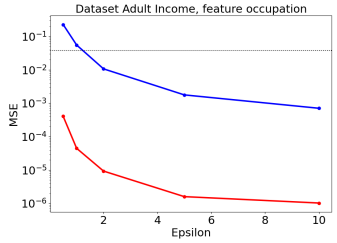

(f) ALE; Census Income; 'hours-per-week'



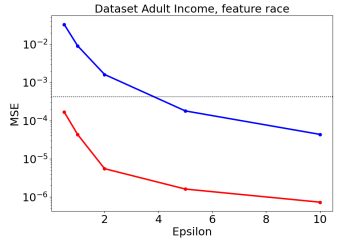

(g) ALE; Census Income; 'marital-status'



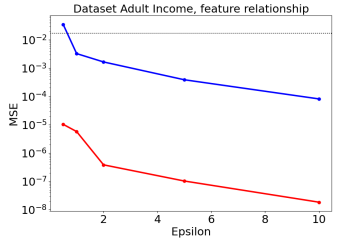

(h) ALE; Census Income; 'native-country'



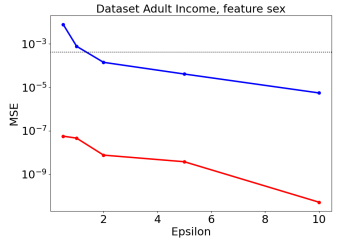

(i) ALE; Census Income; 'occupation'



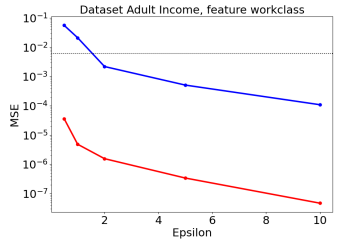

(j) ALE; Census Income; 'race'



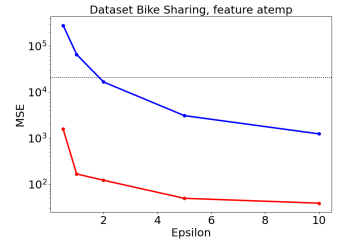

(k) ALE; Census Income; 'relationship'



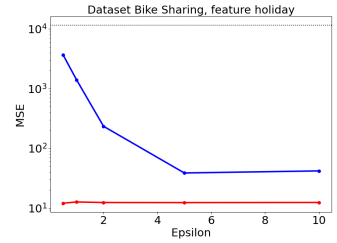

(l) ALE; Census Income; 'sex'




(m) ALE; Census Income; 'workclass'

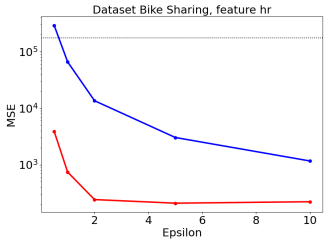



(n) ALE; Bike Sharing; 'atemp'

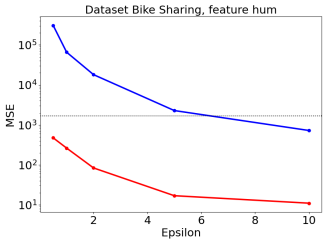



(o) ALE; Bike Sharing; 'holiday'

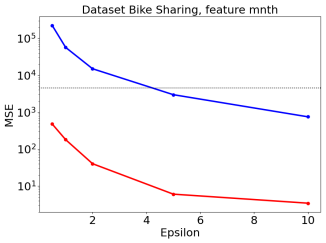
Fig. 14: All results of Experiment 1 for DP ALE and Generic DP ALE.



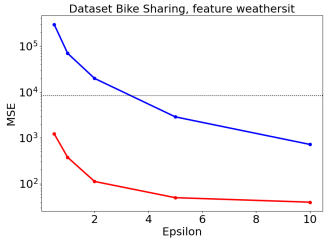

(a) ALE; Bike Sharing; 'hr'



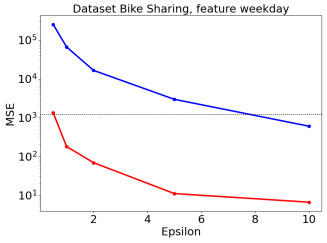

(b) ALE; Bike Sharing; 'hum'



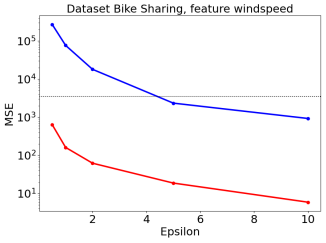

(c) ALE; Bike Sharing; 'mnth'



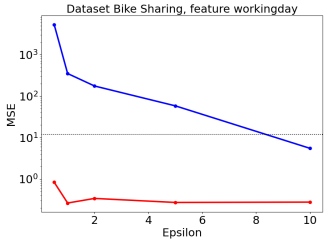

(d) ALE; Bike Sharing; 'weathersit'



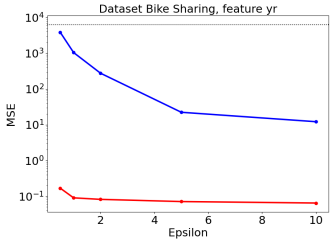

(e) ALE; Bike Sharing; 'weekday'



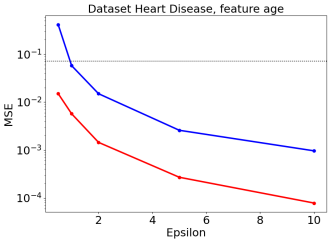

(f) ALE; Bike Sharing; 'windspeed'



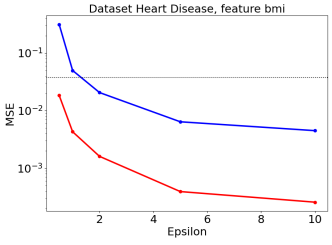

(g) ALE; Bike Sharing; 'workingday'



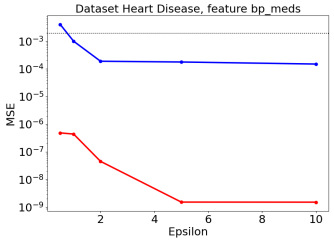

(h) ALE; Bike Sharing; 'yr'



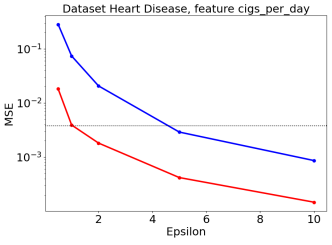

(i) ALE; Heart Disease; 'age'



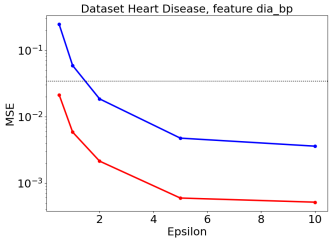

(j) ALE; Heart Disease; 'bmi'



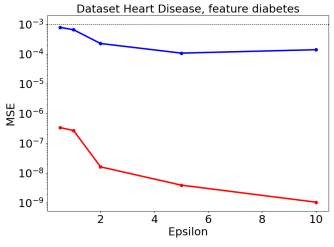

(k) ALE; Heart Disease; 'bp_meds'



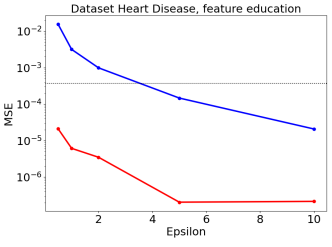

(l) ALE; Heart Disease; 'cigs_per_day'




(m) ALE; Heart Disease; 'dia_bp'

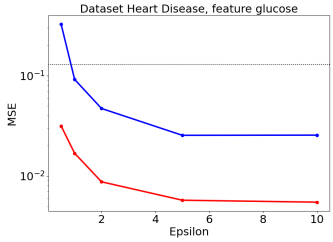



(n) ALE; Heart Disease; 'diabetes'

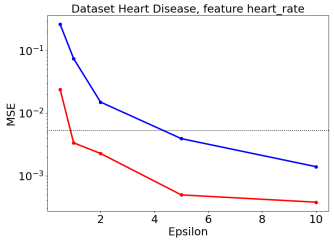



(o) ALE; Heart Disease; 'education'

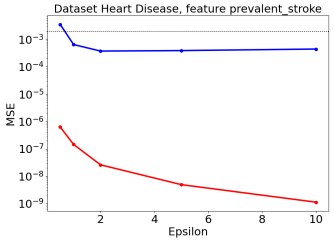
Fig. 15: All results of Experiment 1 for DP ALE and Generic DP ALE.



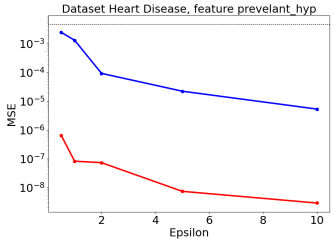

(a) ALE; Heart Disease; 'glucose'



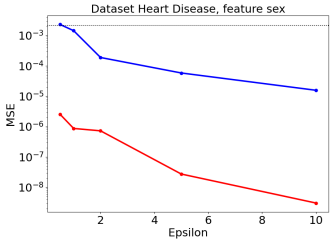

(b) ALE; Heart Disease; 'heart_rate'



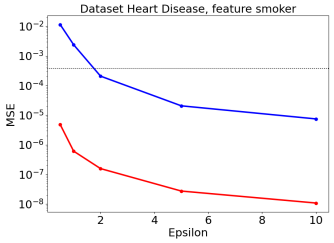

(c) ALE; Heart Disease; 'prevalent_stroke'



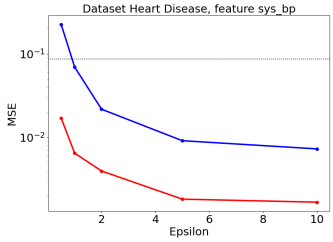

(d) ALE; Heart Disease; 'prevelant_hyp'



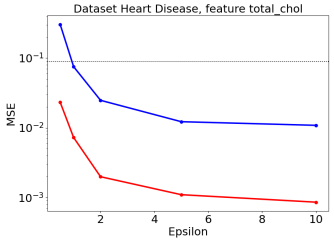

(e) ALE; Heart Disease; 'sex'




(f) ALE; Heart Disease; 'smoker'




(g) ALE; Heart Disease; 'sys_bp'




(h) ALE; Heart Disease; 'total_chol'

Fig. 16: All results of Experiment 1 for DP ALE and Generic DP ALE.